

软件测试黑盒测试

（一）黑盒测试的基本原理与特点

黑盒测试仅需要知道被测对象的输入与预期输出，不需要了解其实现细节

特点：

- ①黑盒测试对测试人员的技术要求相对较低
- ②黑盒测试不需要了解程序的实现细节，因此，测试团队与开发团队可以并行完成各自的任务

缺点：

测试结果覆盖度不容易度量，测试的潜在风险较高

（二）测试方法的评价

典型的黑盒测试方法包括**等价类测试**、**边界值测试**、**基于决策表的测试方法**

从以下几个方面来衡量测试方法的质量：

- 1. 测试用例对被测对象的覆盖率
- 2. 测试用例的冗余
- 3. 测试用例的数量
- 4. 测试用例对缺陷的定位能力
- 5. 测试用例的复杂度

（三）边界值测试

基本原理：在被测对象的边界及边界附近设计测试用例

难点：

- 1. 如何选定输入域或输出域，已进行后续边界的测试用例设计；
- 2. 如何确定输入域或输出域的边界，确保覆盖被测对象所有可能的边界
- 3. 如何确定输入域与输出域边界附近的邻域范围，便于及时发现所有潜伏在边界附近的缺陷
- 4. 如何根据被测对象的边界及其邻域设计测试用例

边界的确定

- 1. 若输入条件规定了取值范围，则该范围可作为边界
- 2. 若输入条件规定了值的个数，则以值的个数为边界
- 3. 若输入域是有序集合（有序表，顺序文件等），则选取结合中特定次序的数据作为边界的第一个或者最后一个数据等

边界点的邻域

边界值附近大小为1个单位长度的范围作为邻域

测试用例的设计

测试数据的选择：

1. 穷举法

2. 典型值法（常用）：针对于某一个边界点 X_{min} ，邻域为 a ，则可取 $X_{min} - a$ ， $X_{min} + a$ 这三个值作为测试数据，测试数据具有典型性且测试用例数量大大减少

边界组合方式的选择

强边界法：覆盖所有输入条件所有边界的组合，对应每个输入条件同时取边界数据的情况

弱边界法：弱边界法的思想基于一个重要的假设——单缺陷假设，无需多个输入条件同时取得边界，由此可知，弱边界法每次仅有一个数据取边界值，其余数据取正常值

全边界法：将所有的边界组合情况都纳入测试用例中，缺点是数据量太过于庞大

设计时应使用**典型值法+弱边界法**

测试用例个数的计算（感觉会考吧）

假设现在有两个输入条件 x, y 且每个输入条件有 n_x, n_y 个边界点（边界点指的是 X_{min} and X_{max} 不包含邻域）：

$$N = 3n_x(n_y - 1) + 3n_y(n_x - 1)$$